УДК 519.688

КОНСТРУКТИВНАЯ МАТЕМАТИКА И ПРИНЦИП БЛИЗКОДЕЙСТВИЯ

© Г. И. Малашонок

Ключевые слова: конструктивная математика, принцип близкодействия, параллельный алгоритм, кластерные вычисления.

Обсуждаются современные направления развития конструктивной математики, которые связаны с появлением многопроцессорных вычислительных систем. Обсуждаются требования к алгоритмам и структурам данных, которые предназначены для многопроцессорных систем.

1 Введение

Причины развития отдельных научных областей – это предмет философии науки. Когда говорят о причинах развития математики последних веков, считается общепризнанным, что внешние побудительные мотивы развития находились в области естествознания и, главным образом, в области решения физических задач, а во второй половине XX в. развитие вычислительной техники стало сильно сказываться на развитии конструктивной математики.

Первые поколения компьютеров, представляющие собой, по-сути, большие арифмометры, привели к развитию такого направления, как численные методы. В большинстве разделов математики появились дополнительные главы, посвященные численным методам. Главная проблема всех численных методов – это устойчивость вычислительных методов, борьба с погрешностью вычислений. С ростом масштабов вычислений детерминистские численные методы почти повсеместно проиграли итерационным численным методам. Появились такие разделы, как теория алгоритмов, теория сложности вычислений.

Следующие поколения компьютеров, позволяющие проводить сложную символьную обработку текстов, привели к появлению символьной компьютерной математики, которую обычно называют компьютерной алгеброй. Выкладки, которые приходится делать студенту, который изучает математику или применяет математический аппарат в прикладных задачах, могут быть осуществлены с помощью компьютера. При этом, с одной стороны, можно избежать механических ошибок, а с другой стороны, можно оперировать очень большими аналитическими выражениями, недоступными без компьютера.

Это привело к развитию многих конструктивных направлений в математике, которые не могли быть интересны в эпоху расцвета численных методов.

Однако аналитические вычисления характеризуются взрывным ростом вычислительной сложности. Поэтому и на персональных компьютерах, и на крупных рабочих станциях удается решать только относительно небольшие задачи, скорее учебного, чем производственного характера.

Современный компьютерный парк представляют вычислительные комплексы, составленные из десятков тысяч параллельно работающих процессоров, имеющих собственную память и средства коммуникации друг с другом и с ведущим процессором.

Такое оборудование потенциально способно решать гигантские аналитические и численно-аналитические задачи. Однако отсутствует теория параллельных алгоритмов.

Параллельные алгоритмы принципиально отличаются от последовательных алгоритмов наличием, кроме собственно вычислительной, еще и коммуникативной составляющей: нужно не только вычислять промежуточные результаты, но и пересылать их от одного процессора к другому согласованным образом. Самые замечательные алгоритмы, которые имеют низкую вычислительную сложность, могут быть совершенно не пригодны для параллельных вычислений.

Рассмотрим два класса алгоритмов, которые играют роль «крайних случаев» для параллельных вычислительных алгоритмов.

Самый замечательный класс параллельных алгоритмов это переборные алгоритмы. Это алгоритмы, которые сводятся к поиску решения путем перебора конечного множества претендентов на решение. Например, путем подстановки в некоторую функцию, которая возвращает результат «истина», если претендент является искомым ответом, и «ложь», если претендент не подходит.

Например, задача поиска минимума функции в некоторой области может решаться таким переборным алгоритмом. Нужно разбить область поиска на много подобластей и искать минимум в каждой подобласти, а затем сравнить полученные значения и выбрать минимальный.

Самый неудобный класс алгоритмов – итерационные алгоритмы. Это алгоритмы, у которых следующая итерация вычислений не может начаться, пока не завершится предыдущая. Например, алгоритм, который вычисляет 10000-й член последовательности, у которой задано 10 первых членов и каждый последующий член определяется предыдущими десятью членами.

2 Матричные алгоритмы и принцип близкодействия

Важный класс алгоритмов составляют конструктивные матричные алгоритмы. Это алгоритмы вычисления матричных функций, когда матрицы имеют большой размер, а элементами матриц являются символьные или численно-символьные выражения.

Классический подход в виде вычисления LU-разложения или применение метода Гаусса не позволяют получить эффективную параллельную программу. Во-первых, на каждом шаге требуется пересылать данные из текущей строки во все остальные строки. Во-вторых, всякий раз, когда необходимо выбирать ведущий элемент, а без этого, как известно, алгоритм Гаусса просто остановится, например, на нулевом ведущем элементе, нужно запускать очень «дорогой» коммуникативный процесс. Результатом такого процесса должен быть выбор нового ведущего элемента, перестроение вычислительного процесса и возобновление процесса вычислений.

Для такой организации требуется некоторое централизованное управление вычислительным процессом. Этапы вычислений сменяются этапами перестройки вычислительного процесса, и чем больше процессоров, тем «дороже» перестройка вычислительного процесса.